MutekH quick start guide for SoCLib platform

This guide explain how to run MutekH on a SoCLib hardware simulator. This is allow easy experimentation with advanced multi-processor programming.

You are **highly encouraged** to first follow the MutekH as Unix process quick start guide which introduce more basic concepts.

MutekH for SoCLib can be compiled for Mips, Arm, PowerPC processors. Other processors are available with different platforms.

The SoCLib platform

The MutekH kernel source code is fully configurable and can be tweaked to adapt hardware platform and application needs. Configuration is handled by a dedicated tool which check dependencies and other relationships between the large set of available configuration tokens.

The example below explains how to setup a SoCLib hardware simulator with 4 RISC processor (Mips, Arm or PowerPC).

Getting SoCLib

We now need to have a working SoCLib install. SoCLib installation is explained here: ?https://www.soclib.fr/trac/dev/wiki/InstallationNotes

SoCLib platform description

The SoCLib source tree contains a platform dedicated to this tutorial: `soclib/soclib/platform/topcells/caba-vgmn-mutekh_tutorial/`.

The MutekH part

Getting the sources

```
svn co -r 1024 https://www-asim.lip6.fr/svn/mutekh/trunk/mutekh
```

Writing the example source code

Note: This example is available directly from `examples/hello` directory in source tree: trunk/mutekh/examples/hello

- Writing the source code in `hello.c`

```
#include <pthread.h>

pthread_mutex_t m;
pthread_t a, b;
```

```
void *f(void *param)
{
  while (1)
    {
      pthread_mutex_lock(&m);
      printf("(%i) %s", cpu_id(), param);
      pthread_mutex_unlock(&m);
      pthread_yield();
    }
}
int main()
{
  pthread_mutex_init(&m, NULL);
  pthread_create(&a, NULL, f, "Hello ");
  pthread_create(&b, NULL, f, "World\n");
}
```

- Writing the `Makefile`

```
objs = hello.o
```

Getting the cross-compilers

You can rely on the `tools/crossgen.mk` script which comes along with MutekH to build some GNU cross-toolchains:

```
$ tools/crossgen.mk
$ tools/crossgen.mk all TARGET=mipsel-unknown-elf
```

Writing the MutekH configuration

The MutekH configuration for the 4 Mips processors platform is in the trunk/mutekh/examples/hello/config_soclib_mipsel file.

Have a look to the BuildSystem page for more information about configuration system and configuration file format. The ?MutekH API reference manual describes all available configuration tokens.

Platform description

The MutekH software uses hardware enumeration to get details about available hardware in the platform, so the `CONFIG_ARCH_DEVICE_TREE` token is defined in the configuration file. It will let the kernel get the platform layout description from a FlattenedDeviceTree which will be built-in.

Therefore, we have to provide the platform description FlattenedDeviceTree and add it to the Makefile to have it compiled in. The `hello/Makefile` file must contain:

```
objs = hello.o platform-mips.o
```

The current FlattenedDeviceTree source file is trunk/mutekh/examples/hello/platform-mips.dts.

Compiling the application along with MutekH

The MutekH kernel and the application may be built out of the source tree.

Change to the SoCLib platform directory and apply the following steps to experiment with out of tree compilation. You have to setup the following variables:

`MUTEKH_DIR`
: Path to MutekH source tree

`APP`
: Path to application source

`CONFIG`
: MutekH configuration file name

```
$ cd soclib/soclib/platform/topcells/caba-vgmn-mutekh_tutorial
$ make MUTEKH_DIR=~/mutekh/ APP=~/mutekh/examples/hello CONFIG=config_soclib_mipsel all
```

This will build the MutekH kernel along with the application. You can still build MutekH separately as explained in the first part. The simulator can then be built using:

```
$ cd soclib/soclib/platform/topcells/caba-vgmn-mutekh_tutorial
$ make system.x
```

Execution

The simulator needs the MutekH executable file name and the processor type and the number of processors of this type:

```
$ ./system.x mutekh/kernel-soclib-mips.out:mips32:4
```

You may want to refer to other articles and documents available from the main page to go further with MutekH.

The ?SoCLib home page provides a livecd image with more advanced examples ready to compile and run. These examples are using older MutekH revisions though.

Other more advanced topics and guides are available from the Main page.